Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.42 – 47

RESEARCH ARTICLE

Energy Efficient Mobile Relaying in Data Intensive Wireless Sensor Networks

Satish Chikkala#, K. Sathi Reddy*

#M.Tech Student, CSE,
Sri Sai Madhavi Institute of Science & Technology,
Mallampudi, Rajahmundry, Andhra Pradesh
csatish1980@gmail.com

*Assistant Professor, CSE,
Sri Sai Madhavi Institute of Science & Technology,
Mallampudi, Rajahmundry, Andhra Pradesh
sathireddy@ymail.com

Abstract— ***WSN is common in different types of application scenarios. It includes a set of sensor nodes deployed over a geographical area to monitor a variety of phenomenon. WSN become increasingly useful in variety critical applications, such as environmental monitoring, smart offices, battlefield surveillance and transportation traffic monitoring. The sensor nodes are tiny and limited in power. Sensor types vary according to the application of WSN. Whatever be the application, the resources such as power, memory and band width are limited. More over, most of the sensors nodes are throw away in nature. Therefore it is vital to consider energy efficiency so as to maximize the life time of the WSN. This paper presents energy efficient mobile relaying in data intensive wireless sensor networks. The concept of mobile relay is that the mobile nodes change their locations so as to minimize the total energy consumed by both wireless transmission and locomotion. The conventional methods, however, do not take into account the energy level, and as a result they do not always prolong the network lifetime.***

Keywords— *Data intensive; Energy; Relay; Routing tree; WSN*

I. INTRODUCTION

The need to monitor and measure various physical phenomena (e.g. temperature, fluid levels, vibration, strain, humidity, acidity, pumps, generators to manufacturing lines, aviation, building maintenance and so forth) is common to many areas including structural engineering, agriculture and forestry, healthcare, logistics and transportation, and military applications. Wired sensor networks have long been used to support such environments and, until recently, wireless sensors have been used only when a wired infrastructure is infeasible, such as in remote and hostile locations. But the cost of installing, terminating, testing, maintaining, trouble-shooting, and upgrading a wired network makes wireless systems potentially attractive alternatives for general scenarios.

A wireless sensor network (WSN)[1][2] consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound , vibration , pressure , humidity, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. Figure 1 shows an example of a wireless sensor network.

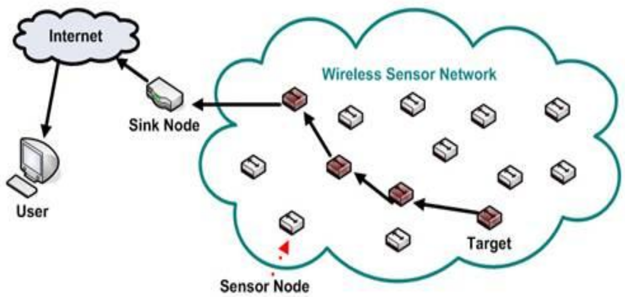

Figure 1: An example of Wireless Sensor Network

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.

Recent advancement in mobile sensor platform technology has been taken into attention that mobile elements are utilized to improve the WSN’s performances such as coverage, connectivity, reliability and energy efficiency. The concept of mobile relay is that the mobile nodes change their locations so as to minimize the total energy consumed by both wireless transmission and locomotion. The conventional methods, however, do not take into account the energy level, and as a result they do not always prolong the network lifetime.

II. Related work

Several different approaches have been proposed to significantly reduce the energy cost of WSNs by using the mobility of nodes. The mobile node may serve as the base station or a “data mule” that transports data between static nodes and the base station [3] [4]. Mobile nodes may also be used as relays [5] that forward data from source nodes to the base station. Several movement strategies for mobile relays have been studied in [5] [6]. Although the effectiveness of mobility in energy conservation is demonstrated by previous studies, the following key issues have not been collectively addressed. First, the movement cost of mobile nodes is not accounted for in the total network energy consumption. Instead, mobile nodes are often assumed to have replenishable energy supplies [7] which is not always feasible due to the constraints of the physical environment. Second, complex motion planning of mobile nodes is often assumed in existing solutions which introduces significant design complexity and manufacturing costs. In [7] [8], mobile nodes need to repeatedly compute optimal motion paths and change their location, their orientation and/or speed of movement. Such capabilities are usually not supported by existing low-cost mobile sensor platforms.

III. Proposed work

In the proposed work, we use low-cost disposable mobile relays to reduce the total energy consumption of data intensive WSNs. Different from mobile base station or data mules, mobile relays do not transport data; instead, they move to different locations and then remain stationary to forward data along the paths from the sources to the base station. Thus, the communication delays can be significantly reduced compared with using mobile sinks or data mules.

A. Energy Optimization Framework

In this section, we formulate the problem of Optimal Mobile Relay Configuration (OMRC) in data-intensive WSNs. Unlike mobile base stations and data mules, our OMRC problem considers the energy consumption of both mobility and transmission. The Optimal Mobile Relay Configuration (OMRC) problem is challenging because of the dependence of the solution on multiple factors such as the routing tree topology and the amount of data transferred through each link. For example, when transferring little data, the optimal configuration is to use only some relay nodes at their original positions.

Assume the network consists of one source si−1, one mobile relay node si and one sink si+1. Let the original position of a node sj be oj = (pj , qj), and let uj = (xj , yj) its final position in configuration U. According to our energy models, the total transmission and movement energy cost incurred by the mobile relay node si is

$$c_i(U) = k\|u_i - o_i\| + am + b\|u_{i+1} - u_i\|^2 m$$

Now We need to compute a position ui for si that minimizes $C_i(U)$ assuming that $u_{i-1} = o_{i-1}$ and $u_{i+1} = o_{i+1}$; that is, node s_i's neighbors remain at the same positions in the final configuration U. We calculate position $u_i = (x_i, y_i)$ for node si by finding the values for x_i and y_i where the partial derivatives of the cost function $C_i(U)$ with respect to x_i and y_i become zero. Position ui will be toward the midpoint of positions u_{i-1} and u_{i+1}. The partial derivatives at xi and yi, respectively are defined as follows:

$$\frac{\delta C_i(U)}{\delta x_i} = -2bm(x_{i+1} - x_i) + 2bm(x_i - x_{i-1}) + k\frac{(x_i - p_i)}{\sqrt{(x_i - p_i)^2 + (y_i - q_i)^2}}$$

$$\frac{\delta C_i(U)}{\delta y_i} = -2bm(y_{i+1} - y_i) + 2bm(y_i - y_{i-1}) + k\frac{(y_i - q_i)}{\sqrt{(x_i - p_i)^2 + (y_i - q_i)^2}}$$

B. Static Tree Construction

We construct the tree for our starting configuration using a shortest path strategy. We first define a weight function w specific to our communication energy model. For each pair of nodes s_i and s_j in the network, we define the weight of edge s_is_j as: $w(s_i, s_j) = a + b\|oi - oj\|^2$ where o_i and o_j are the original positions of nodes s_i and s_j and a and b are the energy parameters. We observe that using this weight function, the optimal tree in a static environment coincides with the shortest path tree rooted at the sink. So we apply Dijkstra’s shortest path algorithm starting at the sink to all the source nodes to obtain our initial topology.

We improve the routing tree by greedily adding nodes to the routing tree exploiting the mobility of the inserted nodes. For each node s_{out} that is not in the tree and each tree edge s_is_j , we compute the reduction (or increase) in the total cost along with the optimal position of s_{out} if s_{out} joins the tree such that data is routed from s_i to s_{out} to s_j instead of directly from s_i to s_j using the LocalPos algorithm described in algorithm 1. We repeatedly insert the outside node with the highest reduction value modifying the topology to include the selected node at its optimal position, though the node will not actually move until the completion of the tree optimization phase. After each node insertion occurs, we compute the reduction in total cost and optimal position for each remaining outside node for the two newly added edges (and remove this information for the edge that no longer exists in the tree). At the end of this step, the topology of the routing tree is fixed and its mobile nodes can start the tree optimization phase to relocate to their optimal positions.

Algorithm 1

function LOCALPOS(o_i, u_i, u_{i-1}, u_{i+1})
  ⊳ *Consider case* si *moves right*
  valid ⟵ FALSE;
  xi ⟵ $\frac{1}{2}$ (xi−1 + xi+1) − Yi;
  if xi > pi **then**
    valid ⟵ TRUE;
  else
    ⊳ *Consider case* si *moves left*
    xi ⟵ $\frac{1}{2}$ (xi−1 + xi+1) + Yi;
    if xi < pi **then**
      valid ⟵ TRUE;
    end if
  end if
  ⊳ *Record if new position is different from previous one*
  if valid **then**
    yi ⟵ $\frac{(xi-1+xi+1-2pi)}{(yi-1+yi+1-2qi)}$ (xi − pi) + qi;
    u′= (xi, yi);
    if ‖u′i – ui ‖ > threshold **then**
      return (u_i, TRUE);
    end if
  end if
  ⊳ *not beneficial to move, stay at original position*
  return (o_i, FALSE);
  end function

C. Tree Optimization Algorithm

In this section, we consider the subproblem of finding the optimal positions of relay nodes for a routing tree given that the topology is fixed. We assume the topology is a directed tree in which the leaves are sources and the root is the sink. We also assume that separate messages cannot be compressed or merged; that is, if two distinct messages of lengths m_1 and m_2 use the same link (s_i, s_j) on the path from a source to a sink, the total number of bits that must traverse link (s_i, s_j) is $m_1 + m_2$. Let the network consists of multiple sources, one relay node and one sink such that data is transmitted from each source to the relay node and then to the sink. We modify our solution as follows. Let s_i be the mobile relay node, S(s_i) the set of source nodes transmitting to si and sdi the sink collecting nodes from si. The cost incurred by s_i in this configuration U is:

$$ci(U) = k\|ui - oi\| + ami + bmi\|ud - ui\|^2$$

Now we obtain the following positions:

$$x_i = p_i + \frac{-Bx(\sqrt{pB^2_x + B^2_y} \pm k)}{A\sqrt{pB^2_x + B^2_y}}$$

$$y_i = q_i + \frac{-By(\sqrt{pB^2_x + B^2_y} \pm k)}{A\sqrt{pB^2_x + B^2_y}}$$

Where

$$A = mi + \sum_{sl \in S(si)} ml$$

$$Bx = mixd + \sum mlxl + Api$$

sl ∈ S(si)

By = miyd + ∑ mlyl + Aqi

sl ∈ S(si)

We note that these values correspond to two candidate points moving in each direction (left/right). The optimal position is the valid value yielding the minimum cost.

Our algorithm starts by an odd/even labeling step followed by a weighting step. To obtain consistent labels for nodes, we start the labeling process from the root using a breadth first traversal of the tree. The root gets labeled as even. Each of its children gets labeled as odd. Each subsequent child is then given the opposite label of its parent. We define mi, the weight of a node s_i, to be the sum of message lengths over all paths passing through s_i. This computation starts from the sources or leaves of our routing tree. Initially, we know $m_i = M_i$ for each source leaf node si. For each intermediate node si, we compute its weight as the sum of the weights of its children. Once each node gets a weight and a label, we start our iterative scheme. In odd iterations j, the algorithm computes a position uj i for each odd-labeled node si that minimizes Ci(Uj) assuming that uj i−1 = uj−1i−1 and uji+1 = uj−1i+1 ; that is, node si's even numbered neighboring nodes remain in place in configuration Uj . In even-numbered iterations, the controller does the same for even-labeled nodes. The algorithm behaves this way because the optimization of u^j_i requires a fixed location for the child nodes and the parent of s_i.By alternating between optimizing for odd and even labeled nodes, the algorithm guarantees that the node s_i is always making progress towards the optimal position u_i. Our iterative algorithm is shown in algorithm 2.

Algorithm 2

```
procedure OPTIMALPOSITIONS(U^0)
   converged ← false;
    j ← 0;
   repeat
         anymove ← false;
         j ← j + 1;
         ⊳ Start an even iteration followed by an odd iteration
         for idx = 2 to 3 do
            for i = idx to n by 2 do
                  (uji ,moved) ← LOCALPOS(oi, S(si), sdi );
                  anymove ← anymove OR moved
                  end for
         end for
         converged ← NOT anymove
   until converged
end procedure
```

Figure 2 shows an example of an optimal configuration for a simple tree with one source node. Nodes start at configuration U^0. In the first iteration, odd nodes (s_3 and s_5) moved to their new positions (u^1_3, u^1_5) computed based on the current location of their (even) neighbors (u^0_2, u^0_4, u^0_6). In the second iteration, only even nodes (s_2 and s_4) moved to their new positions (u^2_2, u^2_4) computed based on the current location of their (odd) neighbors (u^1_1, u^1_3 , u^1_5). Since s_3 and s_5 did not move, their position at the end of this iteration remains the same, so $u^1_3 = u^2_3$ and $u^1_5 = u^2_5$. In this example, nodes did two more sets of iterations, and finally converged to the optimal solution shown by configuration U^6.Even though configurations change with every iteration, nodes only move after the final positions have been computed. So each node follows a straight line to its final destination. As the data size increases, nodes in the optimal configuration get more evenly spaced. In fact, in any given configuration, the maximum distance travelled by a node is bounded by the distance between its starting position and its final position in the evenly spaced configuration.

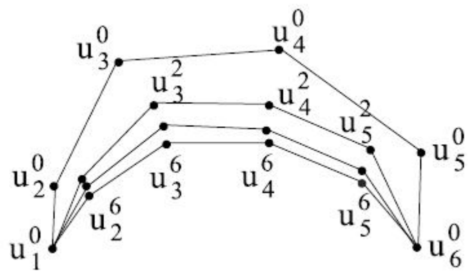

Figure 2: Convergence of iterative approach to the optimal solution

IV. CONCLUSIONS

The main objective of this paper is energy conservation which is holistic in that the total energy consumed by both mobility of relays and wireless transmissions is minimized, which is in contrast to existing mobility approaches that only minimize the transmission energy consumption. The tradeoff in energy consumption between mobility and transmission is exploited by configuring the positions of mobile relays. We develop two algorithms that iteratively refine the configuration of mobile relays. The first improves the tree topology by adding new nodes. It is not guaranteed to find the optimal topology. The second improves the routing tree by relocating nodes without changing the tree topology. It converges to the optimal node positions for the given topology. Our algorithms have efficient distributed implementations that require only limited, localized synchronization.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] Shio Kumar Singh, M.P. Singh, and D.K. Singh, “Applications, Classifications, and Selections of Routing Protocols for Wireless Sensor Networks” International Journal of Advanced Engineering Sciences and Technologies (IJAEST), November 2010, vol. 1, issue no. 2, pp. 85-95.

[3] D. Jea, A. A. Somasundara, and M. B. Srivastava, “Multiple controlled mobile elements (data mules) for data collection in sensor networks,” in DCOSS, 2005.

[4] R. Shah, S. Roy, S. Jain, and W. Brunette, “Data mules: Modeling a three-tier architecture for sparse sensor networks,” in IEEE SNPA Workshop, 2003.

[5] W. Wang, V. Srinivasan, and K.-C. Chua, “Using mobile relays to prolong the lifetime of wireless sensor networks,” in MobiCom, 2005.

[6] D. K. Goldenberg, J. Lin, and A. S. Morse, “Towards mobility as a network control primitive,” in MobiHoc, 2004, pp. 163–174.

[7] A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, “Controllably mobile infrastructure for low energy embedded networks,” IEEE Transactions on Mobile Computing, vol. 5, pp. 958– 973, 2006.

[8] G. Xing, T. Wang, W. Jia, and M. Li, “Rendezvous design algorithms for wireless sensor networks with a mobile base station,” in MobiHoc, 2008, pp. 231–240.